

高速 CCD 数字相机接口设计的 VHDL 逻辑综合的应用

沈宇键, 郝胜国, 郝志航

(中国科学院长春光学精密机械与物理研究所, 吉林 长春 130022)

摘要: 运用 VHDL 语言完成了高速 CCD 数字相机接口设计的描述。整个接口电路由 Lattice 的 1K 系列 CPLD 构成。VHDL 语言与 CPLD 的结合使用, 使接口电路设计的灵活性大大增强, 某些传统上由硬件电路实现的功能转变为软件参与实现, 设计者能够在电路工作过程中对设计随时进行修改。给出了部分接口设计的 VHDL 源代码描述, 通过逻辑综合优化了设计, 实现了设计的时序仿真, 分析了 VHDL 语言的可综合性问题, 并对 VHDL 语言的逻辑综合中的一些问题做了探讨。

关键词: VHDL; 逻辑综合; CCD 可编程逻辑器件; 布局布线; 时序仿真

中图分类号: TP311.5 **文献标识码:** A

1 引言

随着 CCD 技术的发展, 频率高、数字化的新型 CCD 相机不断出现。CCD 相机输出的数字化, 大大简化了相机与采集系统的接口设计, 使得对相机采集的数据的处理更加容易。虽然 CCD 相机趋于数字化, 但是相机的时钟频率也比以往有了很大的提高, 这又使接口设计出现新的问题。中小规模的 TTL 接口电路已无法满足高速相机的时钟要求。随着大规模可编程逻辑器件的普及, 高速 CCD 数字相机的接口设计越来越趋于可编程技术。

VHDL (VHSIC Hardware Description Language VHSIC 硬件描述语言) 作为 IEEE 标准设计语言的出现与可编程技术有着密不可分的关系。对于复杂的可编程逻辑设计, 为了摆脱传统的门级设计方法, 在更抽象的层次上把握和描述系统电路的设计结构和功能特性。VHDL 正被越来越多的硬件工程师所采用。特别是 EDA 技术的不断发展和完善, 高级逻辑综合优化工具的不断出现, VHDL 在高层次电路的设计方面越来越受到重视。

2 VHDL 的逻辑综合

所谓逻辑综合就是将较高抽象层次的描述自动转换到较低层次描述的一种方法, VHDL 语言只有经过综合后才能够用硬件实现。由于 VHDL

语言本身是基于仿真描述的, 因此许多符合语法规则的语言描述是不能被综合的。例如, VHDL 语言中的延迟语句是典型面向仿真的, 它只能作为测试平台中的激励语句使用, 这种语句在综合时将被忽略。

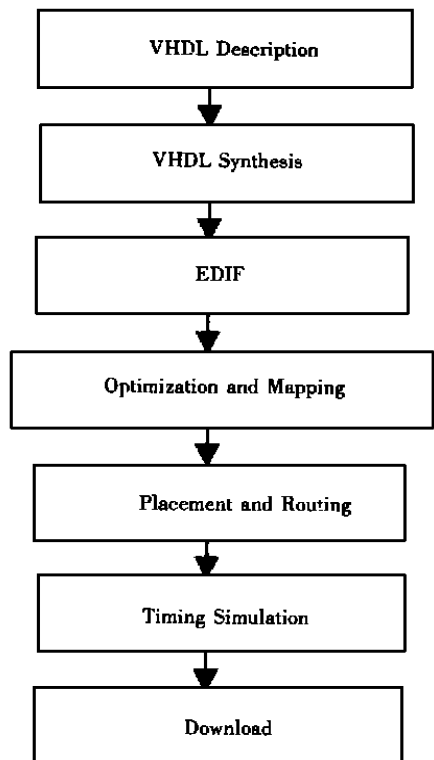


Fig. 1 VHDL design flow

VHDL 语言的设计步骤如图 1 所示, 逻辑综合工具的执行步骤如图 2 所示。

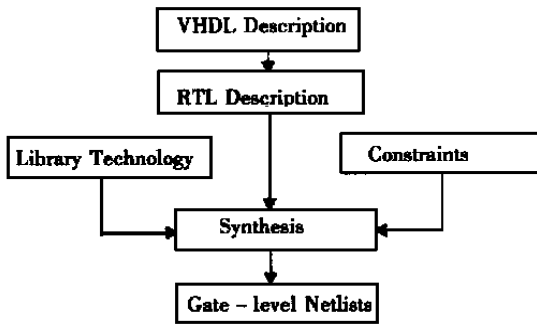


Fig. 2 Synthesis flow

综合工具自动完成将设计的行为级描述 (Behavioral Description) 转换为 RTL (Register Translate Level) 级描述, 然后将 RTL 描述转换为未优化的门级布尔描述(如与、或门, 触发器, 锁存器等), 再执行优化算法, 产生优化布尔描述, 最后按照目的工艺要求, 采用相应工艺库优化的布

尔描述映射成实际的逻辑门。

当前适用于 VHDL 语言的逻辑综合工具有十几种。设计人员只要正确使用这些工具就可以得到系统的门级网络表。对于系统的硬件设计人员, 并不需要详细地了解逻辑综合的细节, 只要知道逻辑综合工具的使用方法和大概情况就行了。

3 高速 CCD 数字相机接口的 VHDL 描述

3.1 高速 CCD 数字相机接口设计

FTM 1010-15 是飞利浦公司的 1K × 1K 帧转移 CCD 数字相机。该相机的输出信号是经过采样、量化后的数据, 精度为 12bit, 相机的像元时钟频率为 20MHz, 图 3 是相机成像系统结构框图。其中, 虚线框内由 CPLD 或 FPGA 完成。本文只阐述框中接口设计部分。图 4 为 FTM 1010-15 的场、行时序图。

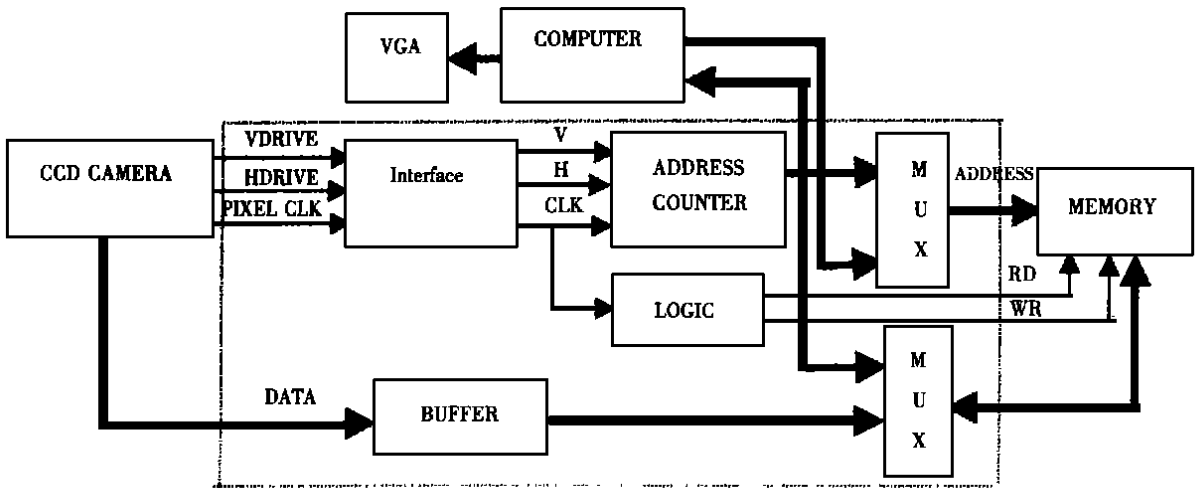


Fig. 3 Block diagram of FT M 1010-15 im age capture system

我们关心的是相机输出的数据与相机行、场及像元时钟的关系。但是从图 4 中可以看到相机输出的像元信号并不都是有用的, 在有效像元前后还存在一些黑电平 (Black) 像元, 哑像元 (Dummy) 以及边缘的过扫描 (Overscan) 像元。同样在一场有效输出前后也存在着一些哑行。在做数据处理前需将这些像元 (行) 去掉, 使每一个有

效像元时钟与每一位有效数据输出能够一一对应。由于相机的像元时钟频率为 20MHz, 一位数据输出的有效周期约为 40ns, 像元时钟的下降沿又位于像元时钟有效后的 12ns (± 5) 处, 因此去掉相机无效像元同时又严格保证像元时序与数据满足相机的时序要求是一个很关键的问题。

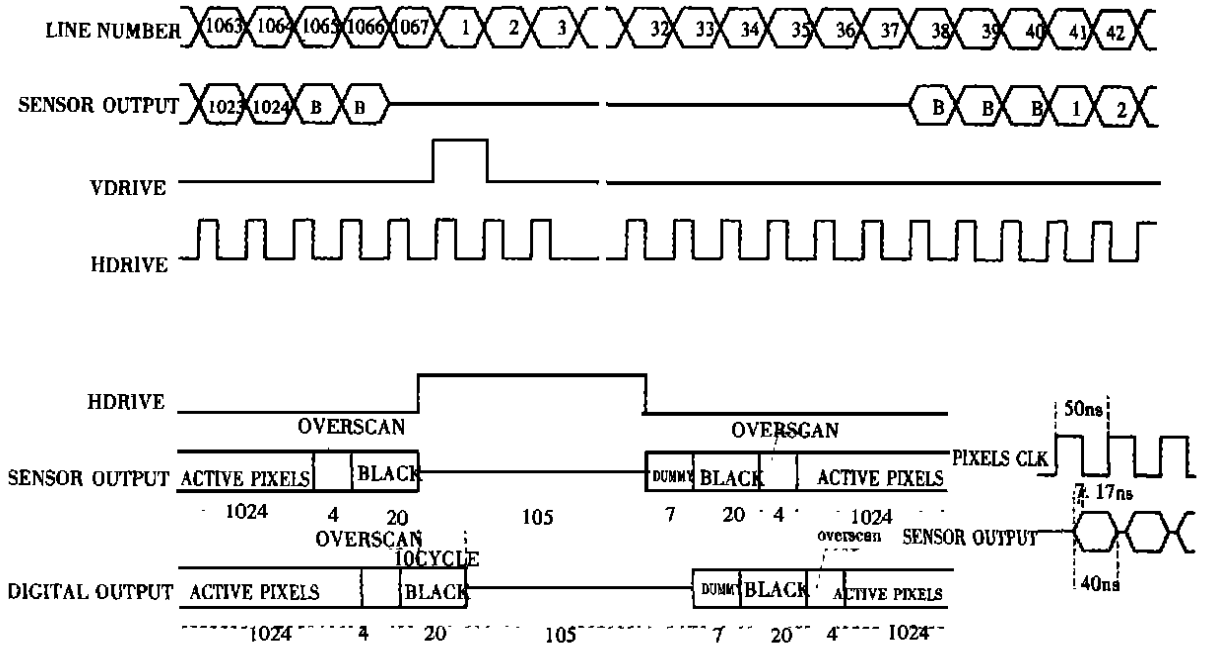


Fig. 4 CCD timing

3.2 VHDL 描述

与之相似, 只是计数值不同而已。

去除无效像元的 VHDL 描述如下, 去除哑行

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY LLL IS
PORT (
    H: IN STD_LOGIC;
    PRESET: IN STD_LOGIC;
    CLK: IN STD_LOGIC;
    BLACKPIXEL: OUT STD_LOGIC_VECTOR (0 TO 7);
    ACTIVEPIXEL: OUT STD_LOGIC_VECTOR (0 TO 9);
    LINE: OUT STD_LOGIC;
    C: OUT STD_LOGIC;
    D: OUT STD_LOGIC;
    CL: OUT STD_LOGIC
);
END LLL;
ARCHITECTURE LLL OF LLL IS
SIGNAL COUNT: STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL COUNT1: STD_LOGIC_VECTOR(9 DOWNTO 0);
SIGNAL CLR: STD_LOGIC;
SIGNAL CLR1: STD_LOGIC;
SIGNAL EN: STD_LOGIC;
SIGNAL E: STD_LOGIC;
BEGIN
BLACKPIXEL <= COUNT;
ACTIVEPIXEL <= COUNT1;

```

----- 端口说明

- 行同步信号, 由相机提供
- 外触发信号, 由外部主程序提供
- 像元时钟信号, 由相机提供
- 有效像元前的 135 个无效像元计数
- 1024 个有效像元计数
- 有效像元计数的清零信号
- 有效像元计数进位信号
- 无效像元的计数进位信号
- 无效像元计数的清零信号

----- 结构描述

----- 局部信号说明

```

CL< = CLR; CL1< = CLR1; D< = EN;
PROCESS(H, PRESET, EN)                                     - - - - - 此进程产生无效像元计数的清零信号
BEGIN
    IF(PRESET= '0' OR EN= '1')THEN
        CLR< = '0';
    ELSIF(H'EVENT AND H= '1')THEN
        CLR< = '1';
    END IF;
END PROCESS;
PROCESS(CLK, CLR)                                         - - - - - 此进程为无效像元计数
BEGIN
    IF(CLR= '0')THEN
        COUNT< = "00000000";
    ELSIF(CLK'EVENT AND CLK= '1')THEN
        IF(COUNT= "10001000")THEN
            COUNT< = "00000000";
        ELSE
            COUNT< = COUNT+ '1';
        END IF;
    END IF;
END PROCESS;
PROCESS(CLK, COUNT)                                       - - - - - 此进程产生无效像元计数进位, 该进位触发有效像元计数
BEGIN
    IF(CLK'EVENT AND CLK= '1')THEN
        IF(COUNT= "10000111")THEN
            EN< = '1';
        ELSE
            EN< = '0';
        END IF;
    END IF;
END PROCESS;
PROCESS(PRESET, EN, E)                                     - - - - - 此进程产生有效像元的行信号
BEGIN
    IF(PRESET= '0' OR E= '1')THEN
        CLR1< = '0';
    ELSIF(EN'EVENT AND EN= '1')THEN
        CLR1< = '1';
    END IF;
END PROCESS;
PROCESS(CLK, CLR1)                                        - - - - - 此进程为有效像元行的计数
BEGIN
    IF(CLR1= '0')THEN
        COUNT1< = "0000000000";
    ELSE IF(CLK'EVENT AND CLK= '1')THEN
        IF(COUNT1= "1111111111")THEN
            COUNT1< = "0000000000";
        ELSE
            COUNT1< = COUNT1+ '1';
        END IF;
    END IF;

```

```

END IF;
END IF;
END PROCESS;
PROCESS(CLK, COUNT1)
BEGIN
    IF(CLK'EVENT AND CLK = '1') THEN
        IF(COUNT1 = "111111110") THEN
            C <= '1'; E <= '1';
        ELSE
            C <= '0'; E <= '0';
        END IF;
    END IF;
END IF;
END PROCESS;
END LLL;

```

----- 此进程产生 1024 行后的进位信号

程序设计采用了多进程(PROCESS)语句方式。这是因为VHDL语言中所有进程语句及进程语句外的语句都是并发执行的,并且与它们出现的先后次序无关,只与进程语句中的敏感量有关。当敏感量变化时,进程被激活。采用这种并行语句结构方式在设计上能够尽量避免信号竞争、冒险的产生。对于频率很高、时序要求严格的CCD接口电路避免出现毛刺现象尤为重要。程序中无效像元的计数进位提前计数值一位产生,它在像元时钟上升沿的同步下会自动向后移一位,这是因为计数输出相对像元时钟有一定的延时,而进位信号既要作为无效像元的反馈清零信号又要作为有效像元计数的使能信号,为了保证信号之间的同步关系以及计数准确就需要进位信号要与第一位有效像元相对应。同时,对无效像元多计数一位以保证有效像元的起始地址是零。采用程序中的描述方式可以实现信号间严格的同步关系而又保证计数值不串位。这比通过硬件电路设计实现更加简单、灵活、可靠。

VHDL描述的具体时序仿真波形如图5。需要指出的是这种时序仿真是对器件进行过综合布线后的时序仿真。实际电路的输出波形与仿真能够完全一致。

3.3 逻辑综合工具的选择

作VHDL设计离不开逻辑综合工具的选择。而逻辑综合工具除了前面提到的将VHDL从行

为级描述转化为门级网表外,还有一个很重要的功能就是将VHDL描述语句实现到有限的逻辑资源中去。通常PLD、CPLD和FPGA器件的逻辑资源是有限的,并具有特定的结构,不是所有的VHDL设计都可以实现到任意选择的结构中去。好的逻辑综合工具可以做到将VHDL描述在逻辑功能上进行综合优化,去掉冗余的逻辑,简化电路,并可以将同一段设计实现到多种器件和结构中,然后再比较它们的结果,以便设计者选择出最适合实现设计目标的器件。此外,除了紧密结合器件结构之外,综合和实现工具之间也应该互相配合。逻辑综合是将VHDL代码编译为逻辑式或网表的过程,而逻辑实现是利用特定的可编程器件结构所进行的优化处理。通常综合后的设计分配到CPLD器件的逻辑块中,并通过中央布线区连接有关的逻辑块和I/O单元。可以说综合和实现之间是一种继承的关系,它们之间需要保持一种良好的配合,保证由综合工具传递到布局布线工具的信息可以用来产生最佳的实现结果。

本文选用的综合工具是Synplify5.14。它除了具备上面提到的逻辑综合工具的特点外,综合速度也是最快的。可编程器件选用LATTICE公司的1K系列。逻辑实现也是LATTICE公司的ISPEXPERT软件。Synplify5.14与ISPEXPERT能够很好的配合。

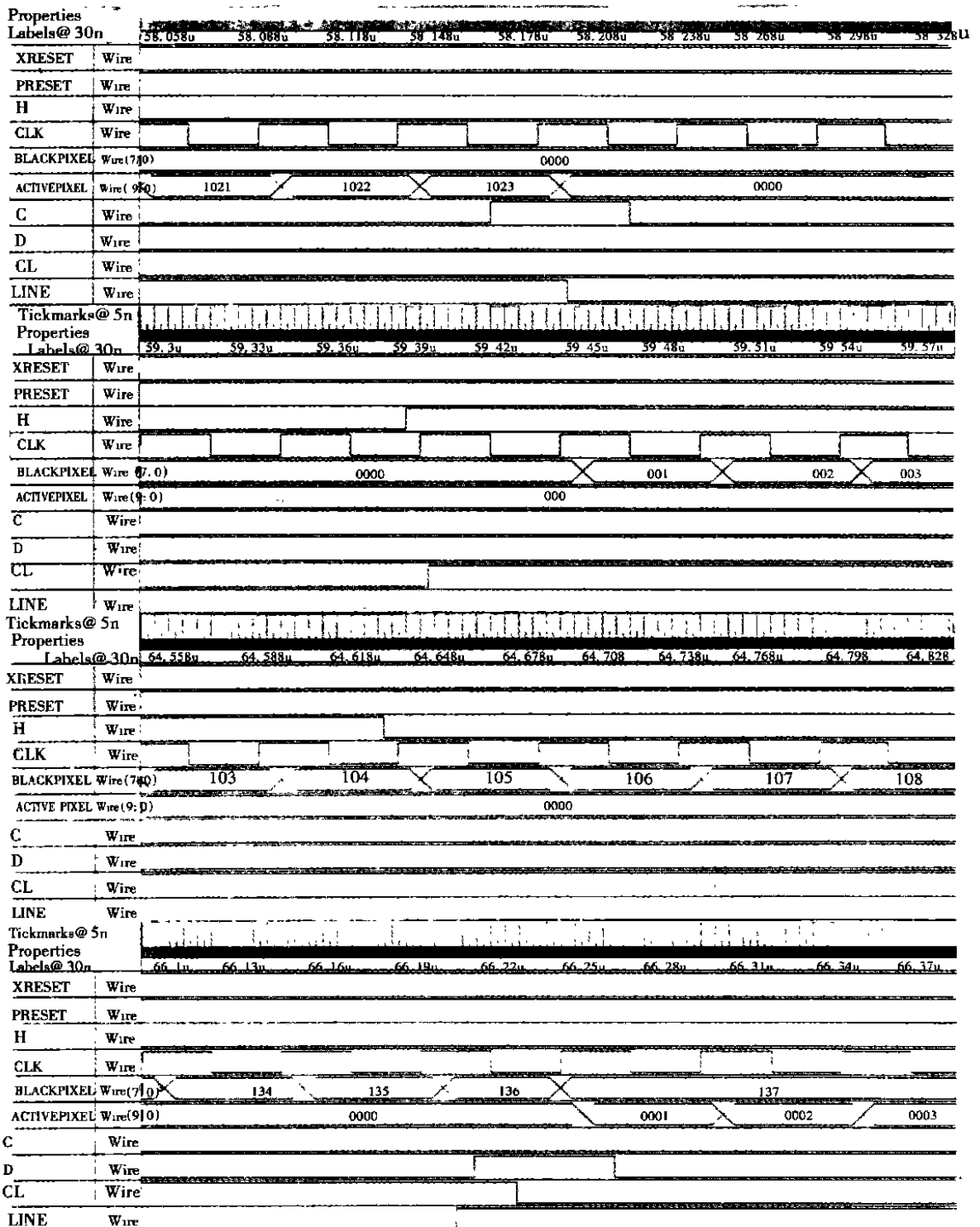


Fig. 5 Timing simulation

4 VHDL 逻辑综合的问题与讨论

VHDL 逻辑综合可以做到对电路进行优化和实现, 但有时为了提高电路速度或某些特殊设计需要而有意设计的程序在做完逻辑综合后往往会被优化掉, 从而得不到预想的结果。为了解决这

一问题, 好的逻辑综合工具在综合 VHDL 时都会提供几种选择命令, 其中有一条是映射(map)命令。该命令是严格按 VHDL 的描述来生成电路, 并不进行逻辑上的优化。但运用这一命令可能造成的后果是器件逻辑资源的紧张。因此在 VHDL 设计时要充分考虑设计要求与可实现的逻辑资源分配之间的关系。另外, 逻辑综合中选用的优化参

数不同,不同的综合命令的运用也会使得实现工具的实际布局布线结果也不相同。为了节约硬件资源和提高系统的性能应当根据设计要求适当的选择优化参数及综合命令。

5 结 束 语

VHDL 语言设计的出现从根本上改变了以往数字电路的设计模式,使电路设计由硬件设计

转变为软件设计。这样提高了设计的灵活性,降低了电路的复杂程度,修改起来也很方便。VHDL 语言虽然功能强大,但存在着可综合性的问题。目前 VHDL 语言的可综合性还没有一个统一的标准,不同厂家提供的逻辑综合工具在综合时,对 VHDL 语言结构的具体要求会有一些细微的差别,随着高级综合系统研究的不断发展,相信这些问题将会得到解决。

参考文献:

- [1] 米良,常青,陈辉煌. VHDL 逻辑综合及 FPGA 实现[J]. 微电子学,1996,26(5): 293- 296.
- [2] 祝常青,张伟功,于伦正. VHDL 语言的可综合性[J]. 微电子学与计算机,1998,4: 25- 29.
- [3] 陈恒. 用 isp- VHDL 设计数字系统[J]. 电子技术,1998,11: 27- 29.
- [4] Kevin Skahill 编著. 可编程逻辑系统的 VHDL 设计技术[M]. 南京: 东南大学出版社,1998.
- [5] FTM 1010D-15 User Manual[M]. Philips Components Image-Optics Modules,1998.

VHDL Logic synthesis and implementation

SHEN Yu-jian, HAO Sheng-guo, HAO Zhi-hang

(*Changchun Institute of Optics, Fine Mechanics and Physics,
Chinese Academy of Sciences, Changchun 130021, China*)

Abstract: Interface design for high-speed digital CCD camera, based on VHDL description, was presented. The interface circuit was made of Lattice 1k series CPLD (Complex Programmable Logic Device). As VHDL and CPLD were used incorporately, some circuit functions traditionally provided by hardware would be involved with software for more flexibility. Designer could modify the design while the circuits were running. Some codes with VHDL for interface circuit were given. VHDL logic synthesis was used to solve the problem of optimum design. Timing simulation of circuit was proposed and simulation results were shown. Synthesizability problems of VHDL design are analyzed and studied.

Key words: VHDL; logic synthesis; CCD FPGA; placement and routing; timing simulation

作者简介: 沈宇键(1972-), 男, 黑龙江呼兰县人, 1996 年于中科院长春光学精密机械研究所获得硕士学位, 现为该所博士研究生。研究方向为数字视频图像处理。